Interactive Story Authoring for Knowledge-Based Support for Investigative Analysis: A Second STAB at Making Sense of VAST Data

Ashok K. Goel, Avik Sinharoy, Summer Adams & Adity Dokania

Design & Intelligence Laboratory,
School of Interactive Computing, Georgia Institute of Technology
Atlanta, GA 30332, USA

Abstract: The sensemaking task in investigative analysis generates stories that connect entities and events in an input stream of data. The Stab system represents crime stories as hierarchical scripts with goals and states. It generates multiple stories as explanatory hypotheses for an input data stream containing interleaved sequences of events, recognizes intent in a specific event sequence, and calculates confidence values for the generated hypotheses. In this report, we describe Stab2, a new interactive version of the knowledge-based Stab system. Stab2 contains a story editor that enables users to enter and edit crime stories. We illustrate Stab2 with examples from the IEEE VAST contest datasets.

Keywords: Investigative Analysis, Sense Making, Story Understanding, Plan Recognition, Hierarchical Scripts, Intelligence Analysis.

This is Georgia Tech GVU Technical Report GIT-GVU-10-02.For additional information please contact the first author at goel@cc.gatech.edu.

Interactive Story Authoring for Knowledge-Based Support for Investigative Analysis: A Second STAB at Making Sense of VAST Data

Ashok K. Goel, Avik Sinharoy, Summer Adams & Adity Dokania

Section 1: Introduction

Intelligence analysis, investigative analysis, and other related forms of information analysis share many common characteristics and components. One unifying element in the various types of information analysis is the task of sensemaking (Bodnar 2005; Heuer 1999; Klein, Moon & Hoffman 2006; Krizan 1999; Pirolli & Card 2005; Thomas & Cook 2005): generation of a model of a situation that connects entities and events in an input stream of data about the situation (sometimes colloquially called the "connect the dots" problem). The input to the sensemaking task in different types of information analysis is characterized by the same kinds of characteristics: the amount of data in the input stream is huge, data comes from multiple sources and in multiple forms, data from various sources may be unreliable and conflicting, data arrives incrementally and is constantly evolving, data may pertain to multiple actors where the actions of the various actors need not be coordinated, the actors may try to hide data about their actions and may even introduce spurious data to hide their actions, data may pertain to novel actors as well as rare or novel actions, and the amount of useful evidence typically is a small fraction of the vast amount of data (the colloquial "needle in the haystack" problem). The desired output of the sensemaking task in different types of information analysis too has the same kinds of characteristics: models that explain the connections among the entities and events, that specify the intent of the various actors, that make verifiable predictions, and that have confidence values associated with them.

Psychological studies of sense making in intelligence analysis (Heuer 1999) indicate that cognitive limitations and biases of human analysts result in several kinds of errors. The three main errors made by human analysts in hypothesis generation are (Heuer 1999): (1) Due to limitations of human memory, analysts may have difficulty keeping track of multiple explanatory hypotheses for a set of data over a long period of time. (2) Due to cognitive fixation, analysts may quickly decide on a single hypothesis for the data set and stick to it even as new data arrives (cognitive fixation). (3) Due to confirmation bias, analysts may look for data that supports the hypothesis on which they are fixated, and not necessarily the data that may refute the hypothesis. Thus, a scientific and technological challenge for cognitive science, artificial intelligence, and human-centered computing is to develop interactive computational tools that can help human analysts overcome these cognitive limitations.

Over the last few years, several researchers have explored the use of knowledge-based techniques to support human decision-making in intelligence and investigative analysis (Birnbaum et al. 2005; Chen at al. 2003; Jarvis, Lunt & Myers 2004; Murdock, Aha & Breslow 2003; PARC 2004; Sanfilippo et. al. 2007; Tecuci et al. 2008; Welty et. al. 2005; Whitaker et. al. 2004). In our previous work, we have developed an automated knowledge-based system called Stab (Adams & Goel 2007a, 2007b, 2008; Goel, Adams, Cutshaw & Sugandh 2009) for IEEE VAST 2006 and 2007 datasets (IEEE VAST 2006; IEEE VAST 2007; Plaisant et al. 2008). Briefly, Stab contains a library of hierarchically organized abstract stories that capture patterns of criminal activity in the IEEE VAST 2006 and 2007 datasets. Thus, in Stab the models that connect the entities and events in the input data are crime stories. The crime stories are represented in the TMKL knowledge representation language (Murdock & Goel 2003, 2008). Stab takes as input a set of events extracted from the IEEE VAST 2006 and 2007 datasets. It gives as output a set of multiple, competing, instantiated stories as explanatory hypotheses that connect the input events and ascribe goals to actors, along with confidence values for each hypothesis as well as a summary of evidence for each hypothesis. Stab's user interface allows the user to view the logical structure of a generated

hypothesis in a graphical representation. It also enables selection of portions of the evidence in support of a generated hypothesis for further analysis of the explanatory relationship between the generated hypothesis and the supporting evidence. A human analyst may use Stab as an external memory of data and explanatory hypotheses. In addition, the analyst may use Stab's results to inform additional search for further evidence for or against a generated hypothesis.

Of course Stab also has several limitations. One of the major limitations of Stab is that crime stories in its story library have to be hand coded in TMKL. This makes it difficult for a user to enter new stories or edit existing stories in Stab's library. To enable a user to more flexibly interact with Stab, we have developed Stab2, a new interactive version of Stab. Stab2 contains a story editor that enables users to enter and edit crime stories in a graphical notation. Stab2 automatically converts a new (or modified) story into its internal knowledge representation language (TMKL). This enables users to interact with Stab's stories "online." In this paper, we first briefly describe the basics of Stab and then present Stab2's story editor.

Section 2: Knowledge-Based Support for Sensemaking

In this section we briefly describe the original Stab system for sensemaking of the IEEE VAST contest datasets. Although Stab's reasoning, knowledge and representations are intended to be general, Stab's library at present contains only crime stories relevant to the VAST-2006 and VAST-2007 datasets.

VAST Datasets: The VAST datasets are synthetic datasets generated by the Pacific Northwest National Laboratories for supporting research and development in the emerging field of visual analytics (Thompson & Cook 2005). The datasets pertain to fictitious illegal and unethical activities, as well as normal and typical activities, in a fictitious town in the United States. The VAST-2006 dataset contains over a thousand news stories written in English, and a score of tables, maps and photographs. Figure 1 illustrates an example news story from the VAST 2006 dataset. The VAST-2007 dataset is a little larger and slightly more complicated but similar in nature.

Inputs to Stab: We manually screened the dataset for stories that indicated an illegal or unethical activity, which left about a hundred news stories out of the more than a thousand originally in the dataset. We then manually extracted events and entities pertaining to illegal/unethical activities. These events/entities form the input to STAB. We also hand crafted representations for each event in terms of the knowledge states it produces. In addition, we examined the maps, photos and tables that are part of the VAST dataset and similarly extracted and represented the relevant information about various entities. Table 1 illustrates a sample of inputs to STAB along with the resulting knowledge state created by an input event.

Torch scandal?

Story by: John Panni
Date Published to Web: 4/30/2004

Political wags in Alderwood are excitedly discussing the impact of steamy photos taken of Mayoral democratic candidate John Torch with an unidentified young brunette woman late one evening at a Tri-Cities Starbucks. Torch, married with 4 children, has not commented on the incriminating pictures. Hawk Press has obtained copies of these pictures, but following company policy, will not publish them.

Incumbent mayor Rex Luther characterized the scandal as "unfortunate". "Moral values are key to anyone wishing to assume a position of leadership and responsibility," he added.

Sources have identified the woman as an employee of Boynton Laboratories. Laurel Sulfate, spokeswoman for the laboratory, was unavailable for comment, currently vacationing in Switzerland. An assistant to Sulfate stated that she "will look into the matter upon her return."

Webmaster
Copyright
Hawk Press Inc.

Figure 1: Example news story from the VAST-2006 dataset.

Sample STAB Inputs	Resulting State
stolen (money $40 Highway-Tire-Store)	Has-object
cured-disease (Boynton-Labs Philip-Boynton prion-disease)	Is-rich-and-famous
named-after (lab Philip-Boynton Dean-USC)	Expert-involved
was-founded (Boynton-Labs)	Is-open
have-developed (Boynton-Labs prion-disease)	Exists-new-disease
announced-investigation (USFDA Boynton-Labs)	Is-investigating
Injected-cow(Boynton-Labs prion-disease)	Cow-is-infected
treatment-cow (Boynton-Labs prion-disease)	Cow-is-cured

(Adapted from (http://www.cs.umd.edu/hcil/VASTcontest06/)
Table 1: A sample of inputs to Stab

Stab's Library of Crime Stories: STAB contains a small library of hierarchically organized abstract stories, or *hierarchical scripts*, that capture the patterns of crimes that occur in the VAST-2006 and VAST-2007 datasets. Figure 2 illustrates a simple script in STAB's library, Rob a Store. This script has the goal of achieving the state of Have Money, given the initial state of Not Have Money (top of figure). This goal (or task), according to the Rob a Store script, is achieved by a plan (or method) that has several actions (or events) in it: Go to Store, Break into Store, Take Money (middle of the figure). Each of these events becomes a task at the next lower level of abstraction in the hierarchical script, and each task can be (potentially) achieved by multiple methods. For example, according to the Rob a Store script, the task of Break into Store can be achieved by Entering through a Window or Entering through a Door (bottom of figure). Each of these methods in turn specifies a process consisting of multiple events, and so on.

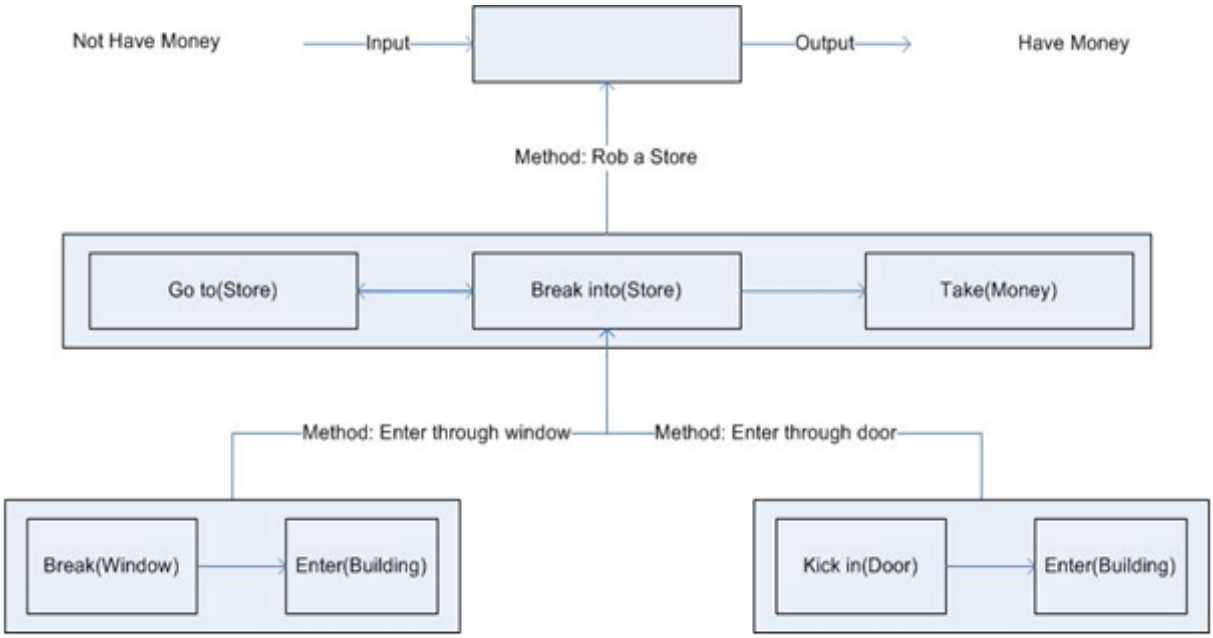

Figure 2: The content and structure of a story in STAB1 (Adapted from Goel et al. 2009)

STAB's hierarchical scripts explicitly represent both goal and state at multiple levels of abstraction. While representation of the state caused by an event is useful for inferring causality, representation of goals of sequences of events is useful for inferring intention. Figure 3 illustrates a more complex script of political conspiracy in which a political figure may get an opponent out of an electoral race either by exposing dirt on him (political blackmail) or having him assassinated. Note this script is composed of several smaller scripts.

We found that seven hierarchical scripts appear to cover all the illegal/unethical activities in the VAST-2006 dataset and that another four hierarchical scripts apparently are sufficient to cover the VAST-2007 as well. We handcrafted this library of scripts into STAB.

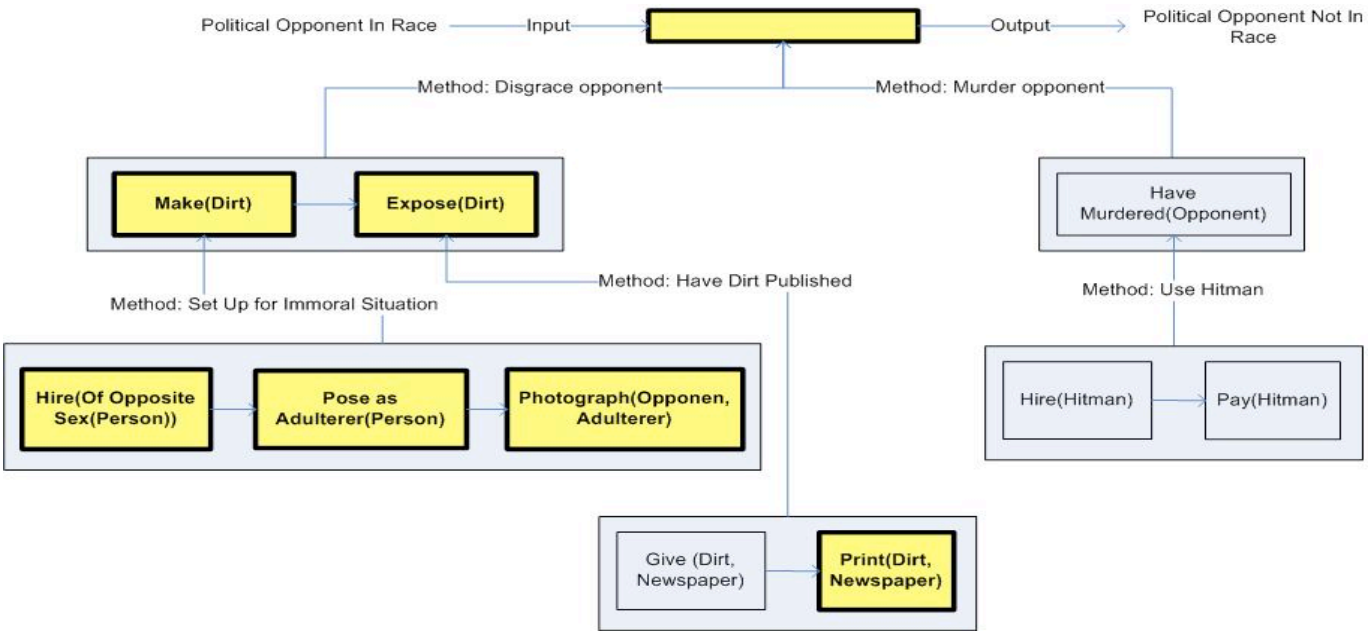

Figure 3: The story for a political conspiracy intended to remove an opponent from an electoral race. Activated nodes are denoted by a thick outline around yellow boxes. (Adapted from Goel et al. 2009)

Stab's Knowledge Representation: STAB's scripts are represented in the TMKL language. A task in TMKL is defined by input knowledge elements, output knowledge elements, required input conditions (pre-conditions), desired output conditions (post-conditions), and methods for implementing the task. A method is defined in terms of subtasks of the task and ordering of the subtasks, and is represented by a finite state machine. Finally, knowledge in TMKL is defined in terms of domain concepts and relationships among them; the input and output knowledge elements in a task specification refer to these domain concepts and relations. Murdock & Goel 2008 provides details of TMKL.

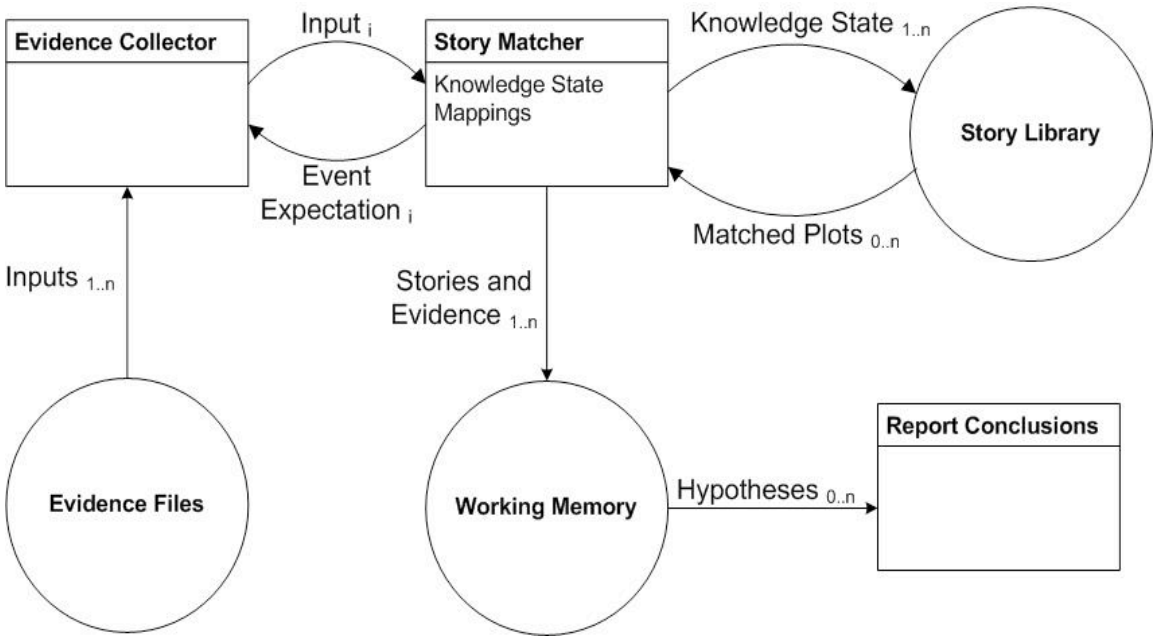

Figure 4: High-Level Architecture of STAB. (Adapted from Goel et al. 2009)

Stab's Computational Architecture: Figure 4 illustrates Stab's computational architecture. First, the Evidence Collector collects the input events in an evidence file in chronological order. Next, the Story Matcher takes one input event at a time and uses its resulting knowledge state of the event with the task nodes in the TMKL representations of the scripts stored in the Story Library. The Story Matcher tags the matching tasks and passes the matching scripts to a Working Memory. This tagging of a matching tasks in effect instantiates the task. The Story Matcher then inspects the next input event in the evidence file and repeats the above process. If the new input event results in the retrieval of a new script, then the script is similarly stored in the Working Memory. If the newly retrieved script is already in the Working Memory, then additional task nodes that match the new input are also tagged but only one script instance is kept.

Stab's Outputs: Stab outputs four kinds of information: all instantiated scripts in the working memory, all nodes in each such script that are matched with an input event, all evidence for each such script, and a confidence value for each such script. The yellow boxes in Figure 4 illustrate the tasks in the political conspiracy script that are matched with an input event in one running of Stab. The confidence value for a script is based on the number of tasks in a script that are matched with input events. Goel et al. (2009) provide details of Stab's reasoning, knowledge, representations and outputs.

Section 3: Interactive Story Editing in Stab2

The interactive Stab2 system is an augmentation of the original Stab system with a primary focus on improving user interaction. The specific goals of Stab2 enhancements included: (i) Allow online authoring and editing of Stab's crime stories in a visual form, and (ii) Enable a user to control all aspects of Stab's reasoning through a graphical interface.

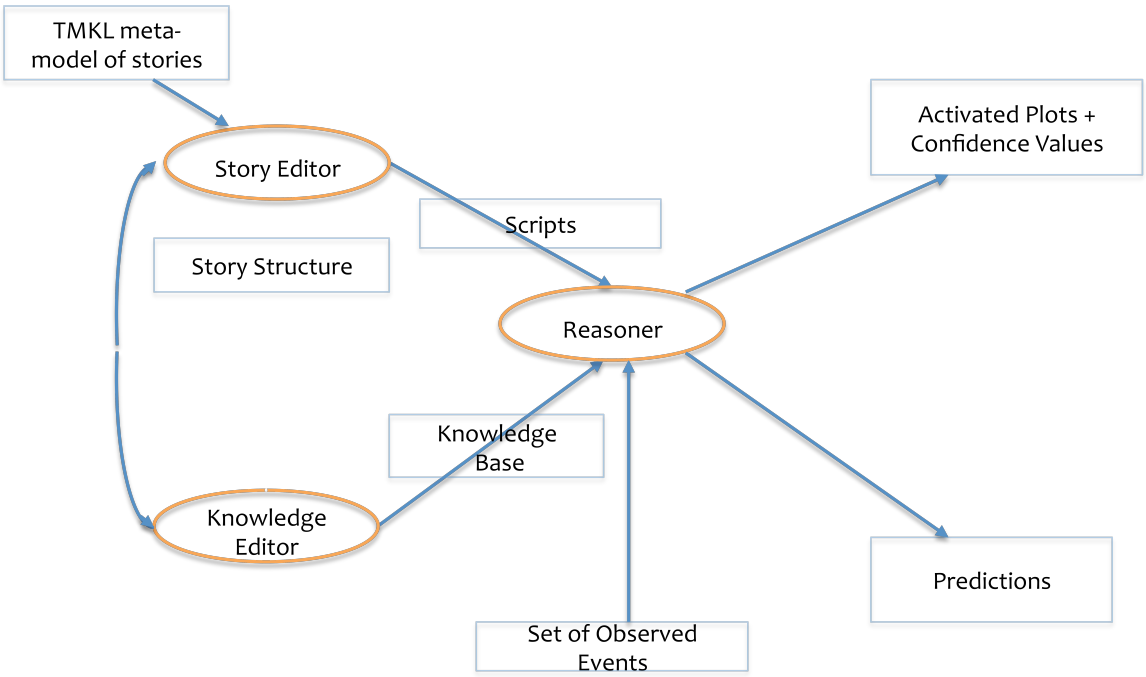

Figure 5: Computational Architecture of the Stab2's Story Authoring Tool

Section 3.1: Interactive Story Authoring in Stab2

Figure 5 illustrates the computational architecture of Stab2's story authoring tool. The Story Editor is used in creating and managing the Stab2's crime stories. These stories are a direct descendent of the hierarchical scripts used in Stab. However, while Stab2's stories still use TMKL as the knowledge representation language, we changed the internal data model in Stab2 to enable automatic translation of a story authored by a user into the TMKL knowledge representation.

The Knowledge Editor too is built into Stab2's interface, and handles user access to the knowledge elements in the TMKL representation of a story. It provides the user control over defining the entities in a story as well as the relations among the entities. The Reasoner in Figure 5 refers to Stab described in the previous section.

Section 3.2: Visual Syntax of Story Author in Stab2

Figure 6 illustrates the visual syntax used by Stab2's story editor. Each task is represented as a rectangle. As the user mouses over a task, the Story Editor displays the TMKL specification of the task. Each method is a represented as a sequence of subtasks that specifies the ordering of the subtasks including iteration. Each subtask in turn is specified as a rectangle with its own methods as the next lower level of abstraction.

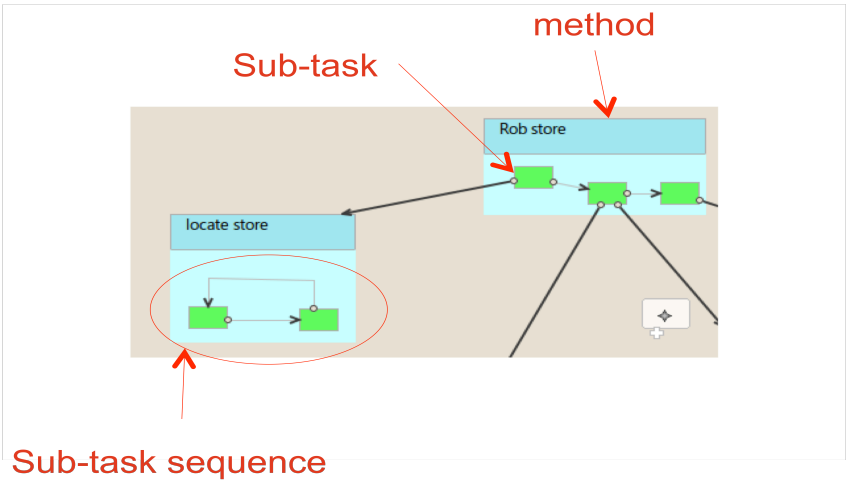

Figure 6: Visual Syntax of the Story Editor in Stab2

Section 3.3: Benefits of Story Editing in Stab2

Figure 7 illustrates Stab2's Story Editor interface. The main part of the figure contains the story being edited, in this case the Rob a Store script illustrated in Figure 2. The palette on the top right of Figure 7 provides the icons for adding new task and method elements to the story. The window on the bottom left depicts the Knowledge Editor. The window on the bottom right indicates other stories in the Story Library and enables the user to browse the library. The graph in the bottom middle enables the user to navigate the script of a story.

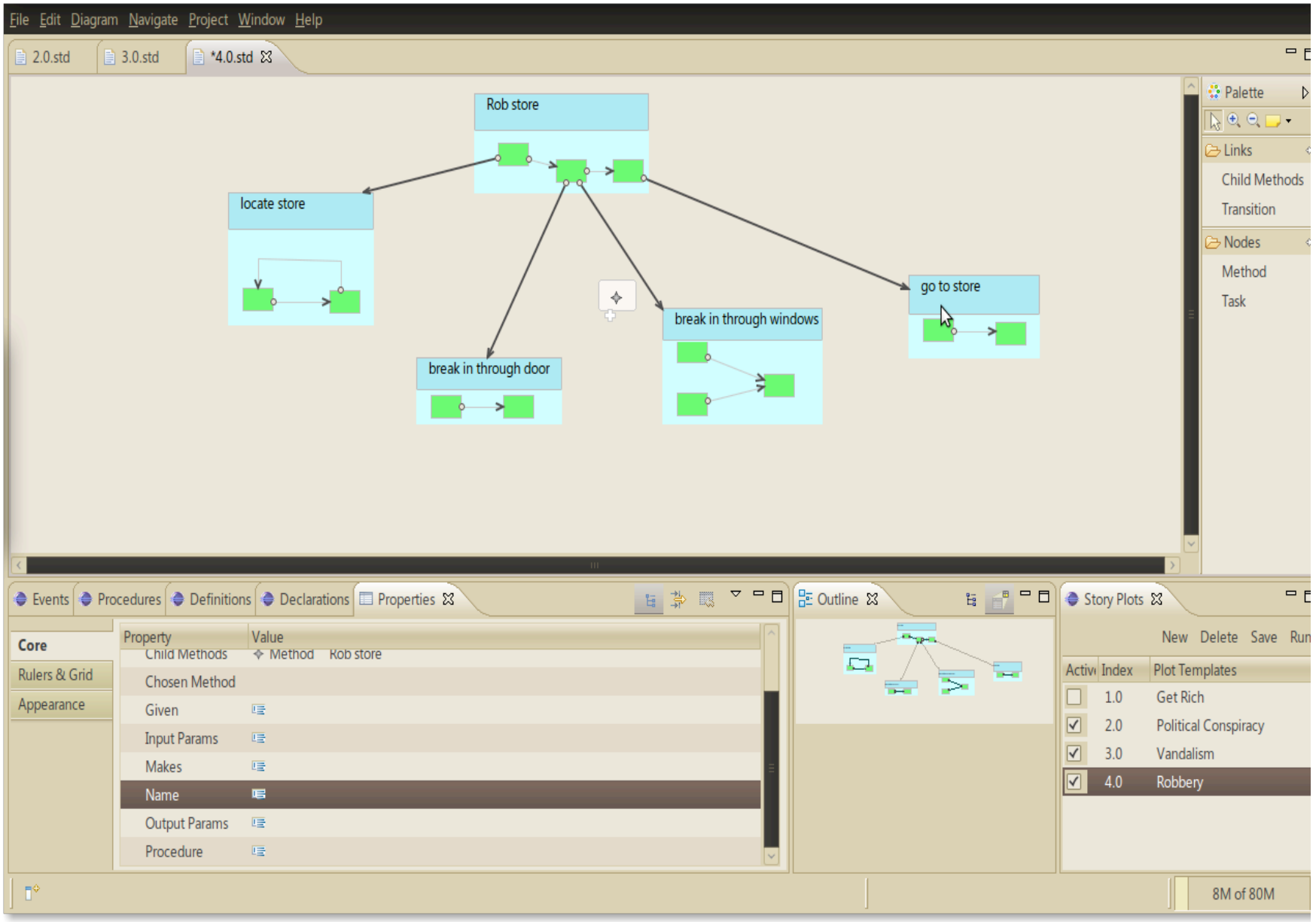

Figure 7: Stab2's Story editor Interface

The Story Editor in Stab2 gives the user more control than was possible in Stab. In the original Stab system, hierarchical scripts were hard-coded in TMKL that itself is coded in the Lisp programming knowledge. This resulted in several difficulties regarding the usability of the system. Firstly, the stored scripts in Stab are not available for reference or review except by perusing the actual Lisp code. This lack of transparency can lead to inaccurate and imprecise understanding of the stories that are generated by Stab as explaining the input data. Stab2's Story Editor provides visual access to both the library of stored scripts as well as the stories instantiated when Stab2 is run for an input dataset.

Secondly, since Stab's stored scripts are coded in TMKL, it is very difficult for a user to author a new script or story, evaluate the correctness of a stored script and make modifications to it. Stab2 Story Editor with its visual interface makes these tasks much easier, though editing the knowledge elements in a story through the Knowledge Editor still is non-trivial.

Finally, the initial version Stab did not allow modifications to the structure of existing scripts at runtime. This implies that it is useful to run Stab just once on an input dataset. However, in general, a human analyst may want to run Stab several times – each with differing variations of the scripts - as the analyst's insight into the criminal patterns evolves. This is perhaps the most important side effect of the use of an interactive Story Editor in Stab2: Stab2 enables an analyst to conduct "what if" simulations. Using Stab2's Story Editor and Knowledge Editor illustrated in Figure 7, an analyst may inspect, modify, activate or deactivate, the scripts stored in Stab2's Story Library, the task and method elements in an individual script, or the knowledge elements like domain concepts and relations. The analyst may then run Stab2 to simulate the effects of the precise conditions she created in Stab2's virtual world.

Section 3.4: Story Authoring in Stab2

When an analyst wants to create a new story, he needs to open the editor and click on add story icon. This opens a white area with a palette on the right. We start with picking the "Main story " icon from the palette and drop it into the white board. This represents the name of the story and is depicted in Figure 8(a). After this we drag the blue rectangle that reads "Intermediary Goals" and connect this to the main story using a Sub-Goal connector all of which can be found in the palette. The result is shown in Figure 8(b). Once the method is created we fill it in with actions performed to achieve that goal. This is done using the "Goals, Action" icon in the palette and this is shown in Figure 8(c). The drag and drop capability makes it very easy for an analyst to construct a new story plot. The tree can go down as many levels depending on the complexity of the story. The relationships between each of the goals are captured using the links. Two kinds of linkages are possible – Sub-Goals that suggests that one goal is a child of another, and Causal Connection that indicates relationships between different actions inside each intermediary goal.

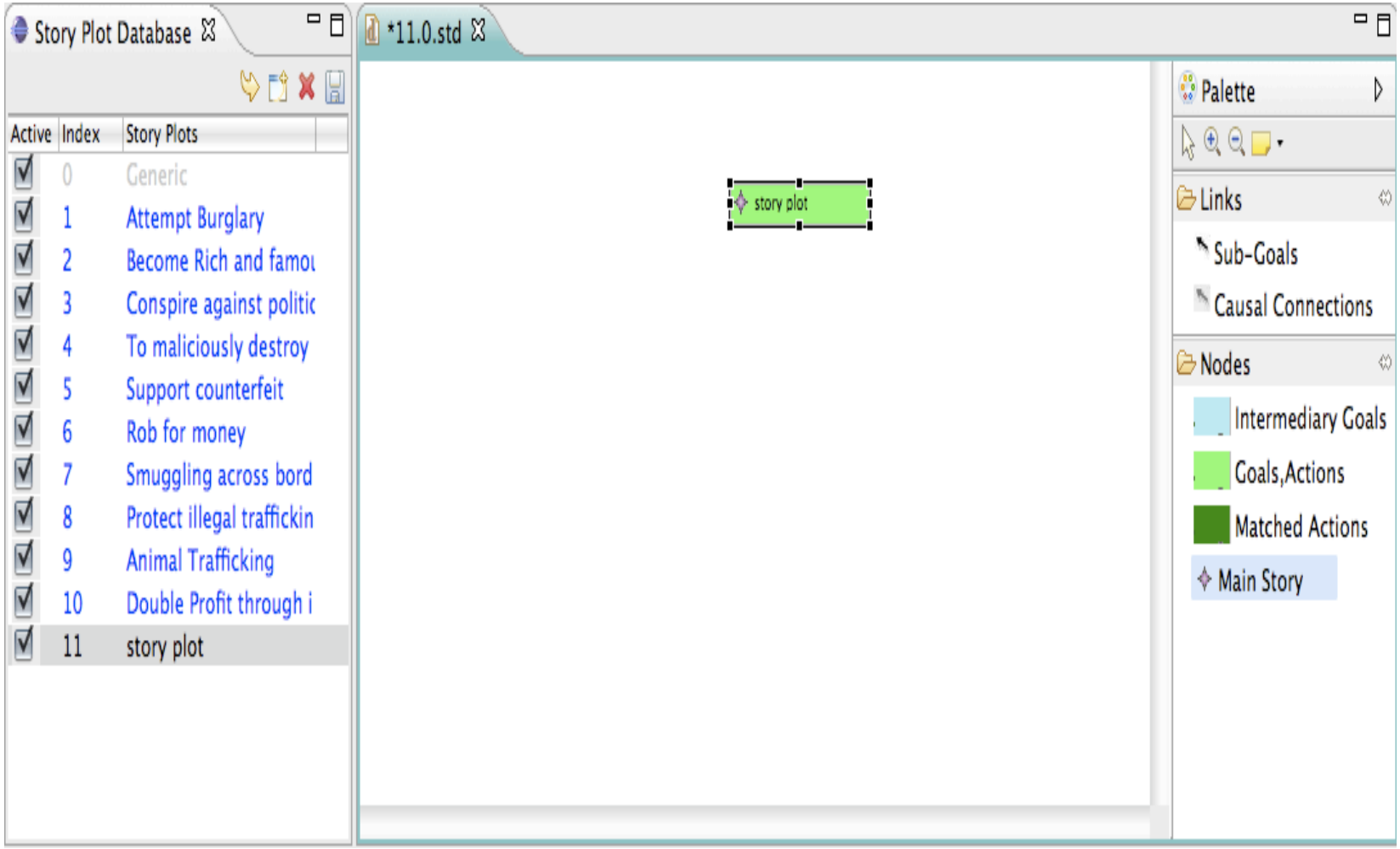

Figure 8(a): 1st Snapshot of Story Authoring in Stab2 – Here the human analyst drags the icon main story from the palette and places it in the editor area and names it as "story plot"

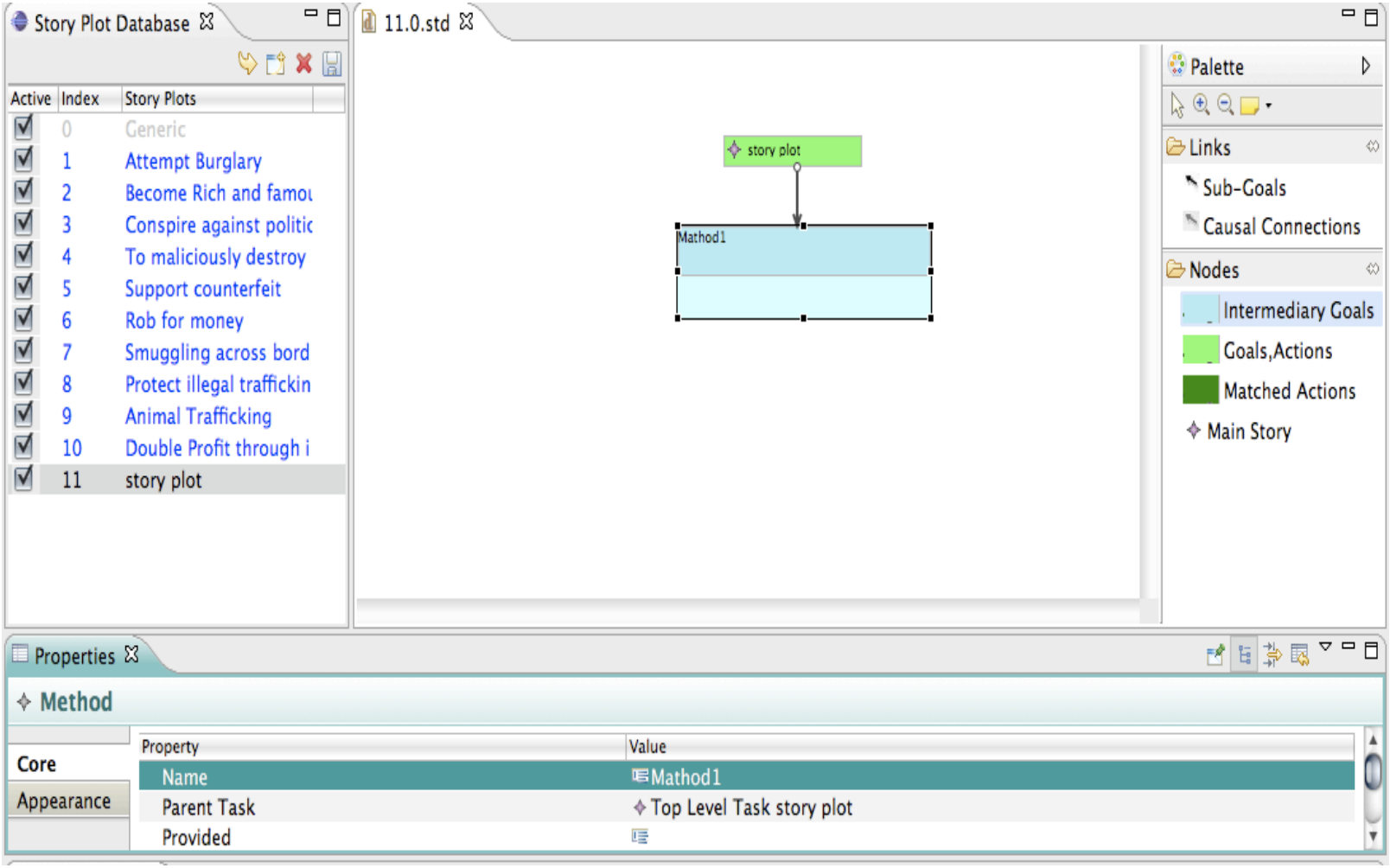

Figure 8(b): A 2nd Snapshot of Story Authoring in Stab2 – Here the human analyst drags the Intermediary Goals icon and places it below the top level task. Also the analyst drags the sub-goals linkage from the palette and connects the two blocks and names the blue rectangle as "Method 1".

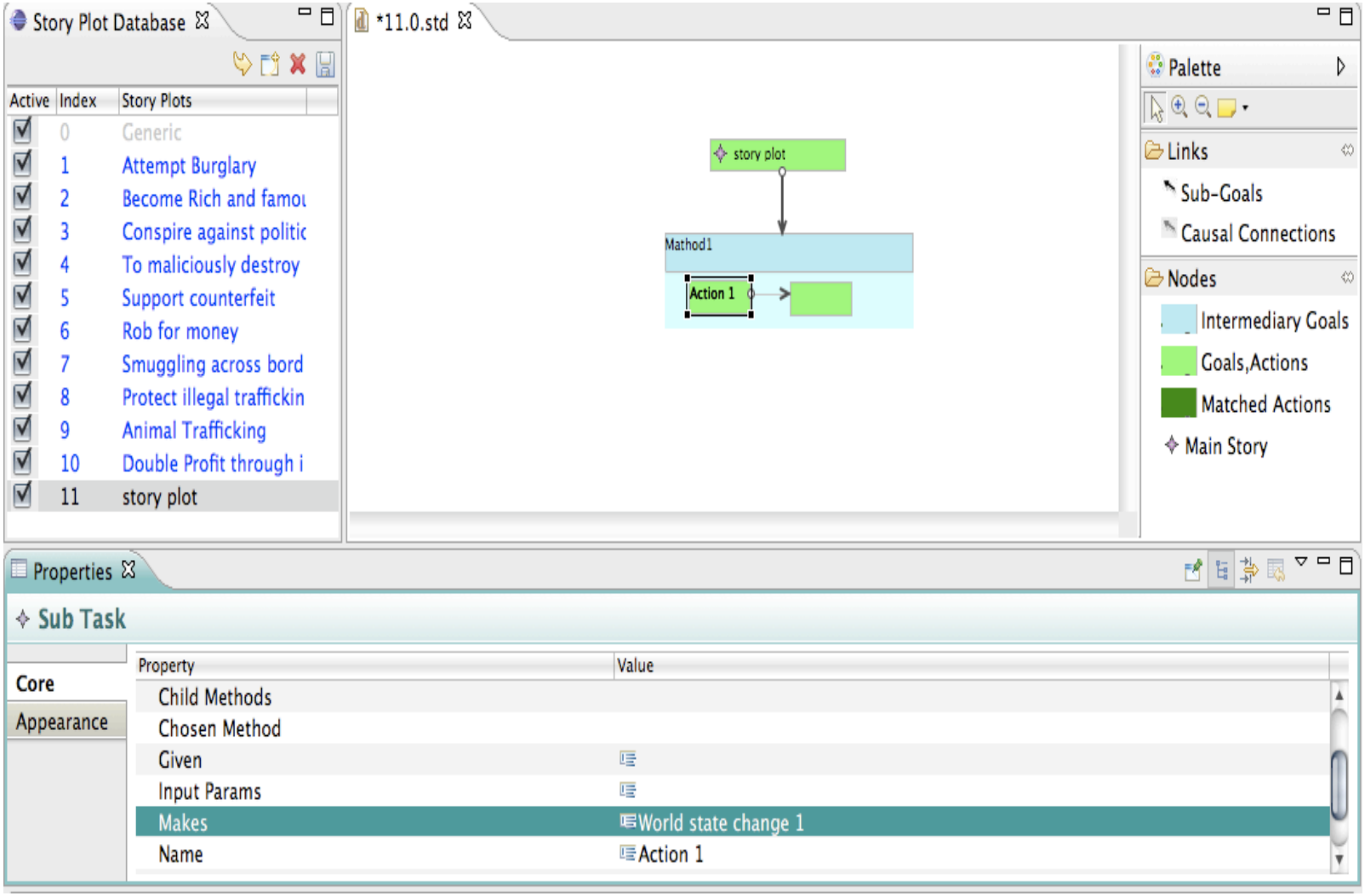

Figure 8(c): A Snapshot of Story Authoring in Stab2 - Here the analyst drags the Goals, Actions icon inside the already created method. He then connects these actions with the causal connections linkage in the palette and names each action.

11.0.std

```
<?xml version="1.0" encoding="UTF-8"?>
<xmi:XMI xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:metamodel="http:///metamodel.ecore" xmlns:notation="http://
www.eclipse.org/gmf/runtime/1.0.2/notation">
  <metamodel:Container xmi:id="_EYWR4YCLEd-6W9eHPog7Tw">
    <allMethods xmi:type="metamodel:Method"
xmi:id="_HD1AsICLEd-6W9eHPog7Tw" name="Method 1"
parentTask="_FXI1YICLEd-6W9eHPog7Tw">
      <childTasks xmi:type="metamodel:SubTask"
xmi:id="_IqrpwICLEd-6W9eHPog7Tw" name="Action 1 " makes="World state
change1" elementId="117"/>
      <childTasks xmi:type="metamodel:SubTask"
xmi:id="_JdpXwICLEd-6W9eHPog7Tw" name="Action 2" makes="World state change
2" elementId="118"/>
      <transitions xmi:type="metamodel:Transition"
xmi:id="_KxV0QICLEd-6W9eHPog7Tw" prevTask="_IqrpwICLEd-6W9eHPog7Tw"
nextTask="_JdpXwICLEd-6W9eHPog7Tw"/>
    </allMethods>
    <topTask xmi:type="metamodel:TopLevelTask"
xmi:id="_FXI1YICLEd-6W9eHPog7Tw" name="story plot"
childMethods="_HD1AsICLEd-6W9eHPog7Tw" elementId="116"/>
  </metamodel:Container>
  <notation:Diagram xmi:id="_EY5EcICLEd-6W9eHPog7Tw" type="Stab"
element="_EYWR4YCLEd-6W9eHPog7Tw" name="11.0.std" measurementUnit="Pixel">
    <children xmi:type="notation:Node" xmi:id="_FXWQwICLEd-6W9eHPog7Tw"
type="1002" element="_FXI1YICLEd-6W9eHPog7Tw">
      <children xmi:type="notation:Node" xmi:id="_FXYF8ICLEd-6W9eHPog7Tw"
type="4002"/>
      <styles xmi:type="notation:ShapeStyle"
```

Figure 9(a): A Snapshot of a XML file for each Story Plot .

Figure 9(b): A Snapshot of a XML file for each Story Plot – This image highlights individual tags. The image on the right highlights the childTask tag and the one the left highlights the allMethods tag.

Each of these stories created are converted into machine understandable language and stored in TMKL format as XML files, which is read and processed during run time. Each story plot looks like Figure 9(a) above. As we can see from the figure each element created in the story has a corresponding tag in the XML document. For example the main story is stored as the topTask tag. The xmi: type of this topTask is metamodel: topLevelTask. The name of this tag is given as "story plot"; this is the name the analyst specified while creating the story. Similarly if you see there is allMethods tag created which is of type metamodel: Method and the name is "Method1" the same as that mentioned by the analyst. Similarly there exists a tag for childTasks. Each Goal, Action has its own childTasks tag and attributes like type, name and makes. One would also notice the transition tag that has attributes such as prevTask, nextTask and helps keep track of causal links between different actions. Each such tag is provided with a machine-generated id to keep track of the relations internally. For e.g.: the topTask tag has an xmi: id –"_FXI1YICLED…"The allMethods tag ends once all its childTasks and transitions in that method have been assigned a tag this can be clearly seen through Figure 9(b).

Related Work

The name "Stab" of our interactive knowledge-based system for sensemaking in investigative analysis comes from STory ABduction. Abduction is inference to the best explanation for a set of data (Bylander et al. 1991; Charniak & McDermott 1985; Fischer et al. 1991; Goel et al. 1995; Josephson & Josephson 1994). We view sensemaking in investigative analysis as constructing a story that explains an input set of data by connecting events through states and ascribing goals to actors.

Stab constructs a story for a given input dataset by retrieving and instantiating hierarchical scripts stored in a library. Scripts, introduced by Schank & Abelson (1977) in AI, have been used in a wide variety of knowledge-based systems especially for story understanding (e.g., Cullingford 1981; Ram 1991; Mueller 2004).

Stab's TMKL knowledge representation language is similar to but more expressive than Hierarchical Task Networks (HTNs) (Erol, Hendler & Nau 1994) for knowledge representation (Lee-Urban 2005). TMKL is more expressive than HTN in part because TMKL enables explicit representation of subgoals and multiple plans for achieving a goal. When Hoang, Lee-Urban and Munoz-Avila (2005) designed a game-playing agent in both TMKL and HTN, they found that "TMKL provides constructs for looping, conditional execution, assignment functions with return values, and other features not found in HTN." They also found that since HTN implicitly provides support for the same features, "translation from TMKL to HTN is always possible."

Stab2's Story Editor and Knowledge Editor enable the authoring and editing of stories represented in TMKL. Interactive story authoring tools have become common in interactive games. Stab2's Story Editor is more similar to interactive story authoring tools used in interactive drama (e.g., Magerko 2005; Mateas & Stern 2005; Riedl & Young 2006).

Although we originally developed TMKL to capture an agent's self-model of its own knowledge, reasoning and architecture (Murdock & Goel 2003, 2008), it has since been used in several other applications. The AHEAD system (Murdock, Aha & Breslow 2003) uses TMKL for representing hypotheses about asymmetric threats. In particular, AHEAD uses past cases of such threats to generate arguments for and against a given hypothesis. The explicit specification of the goals of subsequences of actions allows AHEAD to retrieve past cases relevant to specific subsequences. TIELT (Molineaux & Aha, 2005), an environment for evaluating learning in computer games, uses TMKL as part of its agent description language. In a different project, we are presently using TMKL for specifying the design of game-playing agents (Jones et al. 2009).

Current Work

The story authoring and editing capability of Stab2 enables a human analyst to conduct "what-If" simulations with criminal patterns in order to make sense of the VAST datasets. In its present state of development, however, Stab2 has many limitations. Firstly, the input events to Stab2 at present are extracted from new stories and represented by hand. This is because we have been unable to find an automated tool that can extract events from natural language texts with precision and accuracy. Secondly, the scripts in Stab2 at present are not based on any closed set of primitive actions or tasks. We are exploring the use of Schank's primitive actions (Schank 1983) as the building blocks for our hierarchical scripts represented in TMKL. Thirdly, Stab2 at present does not propagate the values of variables between the nodes in a story. We are augmenting TMKL to enable automatic variable propagation. Fourthly, Stab2 at present does not explain it's reasoning or justifies its conclusions. We are exploring the use of TMKL for supporting self-explanation in Stab2 (Goel et al. 2009; Raja & Goel 2007).

Stab is one of several ongoing research projects at the Southeast Regional Visual Analytics Center (http://srvac.uncc.edu/). These projects include design of interactive techniques and tools for information visualization (e.g., Stasko, Gorg & Liu 2008; Wang et al. 2009), construction of knowledge-based techniques and tools for information foraging (e.g., Liu, Raja, Vaidyanath 2007; Yue et al. 2009), and development of cognitively inspired computational frameworks for visual analytics (e.g., Chang et al. 2009; Green, Ribarsky & Fischer 2008; Liu, Nersessian & Stasko 2008; Ribarsky, Fischer & Pottenger 2009). Our current work on the Stab project is focusing on integrating the interactive knowledge-based system with Jigsaw (Stasko, Gorg & Liu 2008), an interactive tool for visualizing complex relationships among multiple entities.

Acknowledgements

We are grateful to John Stasko for his support and encouragement of this work. This research has benefited from many discussions with him. We thank Jean Scholtz, Jim Thomas and Kristin Cook of the National Visual Analytics Center (NVAC) for their support and encouragement. We also thank William Ribarsky, Anita Raja, Carstern Gorg and James Foley of the Southeastern Regional Visualization and Analytics Center (SRVAC, http://srvac.uncc.edu/) for many discussions on the design and development of Stab2. The description of Stab1 in this report has been adapted from Goel et al. 2009. Our work on the STAB project was sponsored by NVAC under the auspices of SRVAC. NVAC is a U.S. Department of Homeland Security Program led by the Pacific Northwest National Laboratories.

References

Adams, S., & Goel, A. (2007a) STAB: Making Sense of VAST Data, In *Proc. IEEE Conference on Intelligence and Security Informatics*, May 2007.

Adams, S., & Goel, A. (2007b) A STAB at Making Sense of VAST Data. In Proc. AAAI-2007 Workshop on Plan, Activity, and Intent Recognition, Vancouver, Canada, July 2007, pp. 1-8.

Adams, S., Goel, A., & Sugandh, N. (2008) Using AI for Sensemaking in Intelligence Analysis. In Proc. SIAM Datamining Workshop on Link Analysis, Counter Terrorism and Security, Atlanta, April 2008.

Birnbaum, L., Forbus, K., Wagner, E., Baker, J., & Witbrock, M. (2005) Analogy, Intelligent IR, and Knowledge Integration for Intelligence Analysis. In Proc. AAAI Spring Symposium on AI Technologies for Homeland Security, Stanford University, March 2005.

Bodnar, E. (2005) Making Sense of Massive Data by Hypothesis Testing. In *Proceedings of 2005 International Conference on Intelligence Analysis*, May 2005.

Bylander, T., Allemang, D., Tanner, M., & Josephson, J. (1991) The Computational Complexity of Abduction. *Artificial Intelligence,* 49(1-3): 25-60, 1991.

Chang, R., Ziemkiewicz, C., Green, T., & Ribarsky, W. (2009) Defining Insight for Visual Analytics. IEEE Computer Graphics and Applications, 29(2): 14-17.

Charniak, E., & McDermott, D. (1985) *Introduction to Artificial Intelligence*. Reading MA: Addison-Wesley, 1985.

Chen, H., Schroeder, J., Hauck, R., Ridgeway, L., Atabaksh, H., Gupta, H., Boarman, C., Rasmussen, K., & Clements, A. (2003) Coplink Connect: Information and knowledge management for law enforcement. CACM, 46(1):28–34.

Cullingford, R. (1981) SAM and Micro SAM. In R. Schank, & C. Riesbeck (Eds.), *Inside Computer Understanding*: *Five Programs Plus Miniatures.* Hillsdale, NJ: Erlbaum.

Erol, K., Hendler, J., & Nau, D. (1994) HTN Planning: Complexity and Expressivity. In *Proc. Twelfth National Conference on Artificial Intelligence (AAAI-94)*.

Fischer, O., Goel, A., Svirbely, J., & Smith, J. (1991) The Role of Essential Explanations in Abduction. Artificial Intelligence in Medicine, 3(1991): 181-191, 1991.

Goel, A., Josephson, J., Fischer, O., & Sadayappan, P. (1995) Practical Abduction: Characterization, Decomposition and Distribution. *Journal of Experimental and Theoretical Artificial Intelligence,* 7:429-450, 1995.

Goel, A., Adams, A., Cutshaw, N., & Sugandh, N.(2009) Playing Detective: Using AI for Sensemaking in Investigative Analysis. Georgia Tech GVU Technical Report (# GIT-GVU-09-09-03), January 2009.

Goel, A., Morse, E., Raja, A., Scholtz, J., & Stasko, J.(2009) Introspective Self-Explanations for Report Generation in Intelligence Analysis. In Proc. IJCAI-09 Workshop on Explanation-Aware Computing, July 11-12, Pasadena, California.

Green, T., Ribarsky, W., & Fisher, B. (2009) Building and Applying a Human Cognition Model for Visual Analytics.. Information Visualization 8(1): 1-13.

Hoang, H., Lee-Urban, S., & Muñoz-Avila, H. (2005) Hierarchical Plan Representations for Encoding Strategic Game AI. In *Proc. Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE-05).*

Heuer, R. (1999) *Psychology of Intelligence Analysis*. CIA Center for the Study of Intelligence.

IEEE VAST 2007 Contest Dataset. http://www.cs.umd.edu/hcil/VASTcontest06/

IEEE VAST 2007 Contest Dataset. http://www.cs.umd.edu/hcil/VASTcontest07/

Jarvis, P., Lunt, P., & Myers, K. (2004) Identifying Terrorist Activity with AI Plan Recognition Technology. In *Proc. Innovative Applications of AI (IAAI),* pp. 858-863.

Jones, J., Parnin, C., Sinharoy, A., Rugaber, S., & Goel, A. (2009) Adapting Game-Playing Agents to Game Requirements. In Proc. Fifth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE-09), pp. 148-153, Stanford University, California, USA, October, 2009.

Josephson, J., & Josephson, S. (editors). (1994) *Abductive Inference: Computation, Philosophy, and Technology*. Cambridge University Press, 1994.

Klein, G., Moon, B., & Hoffman, R. (2006). Making sense of sensemaking II: A macrocognitive model. *IEEE Intelligent Systems*, 21(5), 88-92.

Krizan, L. (1999) *Intelligence Essentials for Everyone*, Joint Military Intelligence College, 1999.

Lee-Urban, S. (2005). TMK Models to HTNs: Translating Process Models into Hierarchical Task Networks. M.S. Thesis. Department of Computer Science, Lehigh University.

Liu, D., Raja, A., & Vaidyanath, J. (2007): TIBOR: A Resource-bounded Information Foraging Agent for Visual Analytics. In: Proceedings 2007 IEEE/ WIC/ ACM International Conference on Intelligent Agent Technology (IAT 2007), Silicon Valley, CA, Nov 2-5, 2007.

Liu, L., Nersessian, N., & Stasko, J. (2008) Distributed Cognition as a Theoretical Framework for Information Visualization. IEEE Transactions on Visualization and Computer Graphics, Vol. 14, No. 6, November/December 2008, pp. 1173-1180.

Magerko, B. (2005) Story representation and interactive drama. In Proc. 1st Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE05), Marina del Rey.

Mateas, M., & Stern, A. (2005) Structuring content in the Façade interactive drama architecture. In Procs. 1st Artificial Intelligence and Interactive Digital Entertainment (AIIDE05), Marina del Rey.

Molineaux, M., & Aha, D. (2005) TIELT: A Testbed for Gaming Environments. In *Proc. 2005 National Conference on AI (AAAI 2005),* pp. 1690-1691.

Mueller, E. (2004) Understanding Script-Based Stories using Commonsense Reasoning. *Cognitive Systems Research,* 5(4), 307-340.

Murdock, J., Aha, D., & Breslow, L. (2003) Assessing Elaborated Hypotheses: An Interpretive Case-Based Reasoning Approach. In *Proc. Fifth International Conference on Case-Based Reasoning.* Trondheim, Norway, June 2003.

Murdock, J., & Goel, A. (2003) Localizing Planning using Functional Process Models. In Proc. International Conference on Automated Planning and Scheduling (ICAPS-03), June 2003, pp:73-81.

Murdock, J., & Goel, A. (2008) Meta-Case-Based Reasoning: Self-Improvement through Self-Understanding. Journal of Experimental and Theoretical Artificial Intelligence, 20(1): 1-36.

PARC AI Team & Heuer, R. (2004): ACH: A Tool for Analyzing Competing Hypotheses. PARC Technical Report, October 2004.

Pirolli, P., & Card, S. (2005) The Sensemaking Process and Leverage Points for Analyst Technology as Identified Through Cognitive Task Analysis. In *Proceedings of 2005 International Conference on Intelligence Analysis*, May 2005.

Plaisant, C., Grinstein, G., Scholtz, J., Whiting, M., O'Connell, T., Laskowski, S., Chien, L., Tat, A., Wright, W., Gorg, C., Liu, Z., Parekh, N., Singhal, K., Stasko, J. (January 2008) *Evaluating Visual Analytics: The 2007 Visual Analytics Science and Technology Symposium Contest.* IEEE Computer Graphics and Applications. Vol. 28, No. 2, March/April 2008, pp. 12-21.

Raja, A., & Goel, A. (2007): Introspective Self-Explanation in Analytical Agents. In: Proceedings of AAMAS 2007 Workshop on Metareasoning in Agent-based Systems. pp 76-91, Hawaii, May 2007.

Ram, A. (1991) A theory of questions and question asking. The Journal of the Learning Sciences, 1(3-4): 273-318.

Ribarsky, W., Fisher, B., & Pottenger, W. (2009) The Science of Analytical Reasoning. Information Visualization 8(4): 252-262.

Riedl, M., & Young, R. (From linear story generation to branching story graphs. IEEE Computer Graphics and Applications, Special Issue on Interactive Narrative, 26(3), 2006.

Sanfilippo, A., Cowell, A., Tratz, S., Boek, A., Cowell, A., Posse, C., & Pouchard, L. (2007) Content Analysis for Proactive Intelligence: Marshaling Frame Evidence. In *Proc. AAAI 2007*, pp. 919-924.

Schank, R. (1983) Dynamic Memory: A Theory of Reminding and Learning in Computers and People. Cambridge University Press.

Schank, R., & Abelson, R. (1977) *Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures.* Hillsdale, NJ: Erlbaum.

Stasko, J., Gorg, C., & Liu, Z. (2008): Jigsaw: Supporting Investigative Analysis through Interactive Visualization. Information Visualization. Vol. 7, No. 2, Summer 2008, pp. 118-132.

Tecuci, G., Boicu, M., Marcu, D., Boicu, C., & Barbulescu, M. (July 2008): Disciple-LTA: Learning, Tutoring and Analytic Assistance. Journal of Intelligence Community Research and Development (JICRD).

Thomas, J. & Cook, K. (2005) Illuminating the Path. IEEE Computer Society.

Wang, X., Jeong, D., Dou, W., Lee, S., Ribarsky, W., & Chang, R. (2009) Defining and applying knowledge conversion processes to a visual analytics system. Computers & Graphics, 33(5): 616-623.

Welty, C., Murdock, J., Pinheiro da Silva, P., McGuinness, D., Ferrucci, D., & Fikes, R. (2005) Tracking Information Extraction from Intelligence Documents. In *Proceedings of the 2005 International Conference on Intelligence Analysis*, McLean, VA.

Whitaker, E., Simpson, R., Burkhart, L., MacTavish, R., & Lobb, C. (2004). Reusing Intelligence Analysts' Search Plans. In *Proc. Annual Meeting of the Human Factors and Ergonomics*, pp. 367-370.

Yue, J., Raja, A., Liu, D., Wang, X., & Ribarsky, W. (2009): A Blackboard-based Approach towards Predictive Analytics. In Proc. AAAI Spring Symposium on Technosocial Predictive Analytics.